Installation Manual
TCI xPDM Integration
TCI ENOVIA Client extension

PDM Integration

Version	2.2
Last revised	2014-12-10
Status	in progress
Author	Christoph Schlameuß

Protection category: Public

Imprint

Issued by

T-Systems International GmbH
Systems Integration
Fasanenweg 5, 70771 Leinfelden-Echterdingen

File name	Document number	Document name
InstallationManual.docm	not applicable	Installation Manual
Version	**Last revised**	**Status**
2.2	2014-12-10	in progress
Author	**Content checked by**	**Released by**
Christoph Schlameuß	Ulrich Mehringer	Christoph Schlameuß
Contacts	**Telephone / fax**	**E-Mail**
T-Systems International GmbH Product & Process Creation Solutions	+49 (0) 711 999 – 6472	mail: cmi_support@t-systems.com

Brief details

Installation instructions for the T-Systems TCI.

This guide provides installation and configuration information for the ENOVIA V6 Teamcenter Integration. Before using this guide, be sure you understand:

- the administration of the ENOVIA V6 system
- the administration of the Teamcenter system

ENOVIA V6 is a registered trademark of Dassault Systèmes.

Teamcenter is a registered trademark of Siemens PLM Corporation.

Oracle is a registered trademark of Oracle Corporation.

Names of other products mentioned in this manual are used only for identification purpose and may be trademarks of their companies.

Table of Contents

1 Overview 7
1.1 System and Software Requirements 7
1.2 Shipment 7
2 Infrastructure Overview 8
2.1 Overview 8
2.2 Infrastructure Remarks 8
3 Installing the TCI XpdmAdapter 9
3.1 Prerequisites 9
3.1.1 Teamcenter 9
3.1.2 Teamcenter SOA Client 9
3.1.3 ENOVIA 9
3.2 Adapter setup 9
3.2.1 Setup as web application 10
3.3 Adapter configuration 11
3.3.1 Database configuration 11
3.3.2 Logging configuration 12
3.3.3 General adapter configuration 12
3.3.4 Event Configuration 22
3.3.5 V6 Environment File 22
3.3.6 ENOVIA Custom Configuration 23
3.4 Trouble shooting 24
4 TcIC compatibility 26
5 Installing the license manager 27
5.1 Remarks 27
5.2 Windows 7 27
5.3 Running Licman as a regular executable on Windows 27
6 Installing the TCI Teamcenter plug-in 28
6.1 Installation 28
7 Installing the CATIA V6 client extensions 29
7.1 Client extensions configuration 30
A Event flow 31

List of Sources 33
Change History / Release Notes 33

List of Figures

Figure 2-1 Infrastructure Overview 8
Figure 7-1 Installation directory of the CATIA V6 client extensions 29
Figure A.1: Interactive send – event flow 31

List of Tables

Table 3.1: Teamcenter settings. 9
Table 3.2: ENOVIA settings. 9
Table 3.3: TCI web application environment variables 10
Table 6.1: TCI Teamcenter plug-in file to be installed 28
Table 6.2: Teamcenter plug-in - environment variables 28

1 Overview

1.1 System and Software Requirements

Teamcenter Adapter Installation on the following operation systems and software prerequisites:

- Windows Server 2008 R2
- Oracle Java JDK 7 Update 55 Windows x64
- Apache Tomcat 7.0.57

The following remote systems are supported:

- ENOVIA V6R2013X FP1336 or higher (with X-PDM infrastructure)
- 3DEXPERICENCE 2014X[1]
- Teamcenter 2007.1 (4-tier)[2]
- Teamcenter 8.3.0 (4-tier)
- Teamcenter 9.1.0 (4-tier)
- Teamcenter 10.1.0 (4-tier)

The following DBMS are supported:

- Oracle Database 11g Release 2
- Apache Derby DB (JavaDB) 10.6

1.2 Shipment

The software will be delivered on a CD-ROM in ISO-9660 format containing the following parts (depending on desired operating system architecture):

- TCI XpdmAdapter
- TCI Teamcenter plug-in
- TCI Teamcenter template[3]
- TCI CATIA V6 client extensions
- Licman20
- Product documentation

The product makes use of 3rd Party and Open Source Software. A list of the used products and its licenses is distributed with the adapter.

---

[1] Experimental

[2] Server components only (no TCI Teamcenter Plug-in)

[3] Installation optional

2 Infrastructure Overview

2.1 Overview

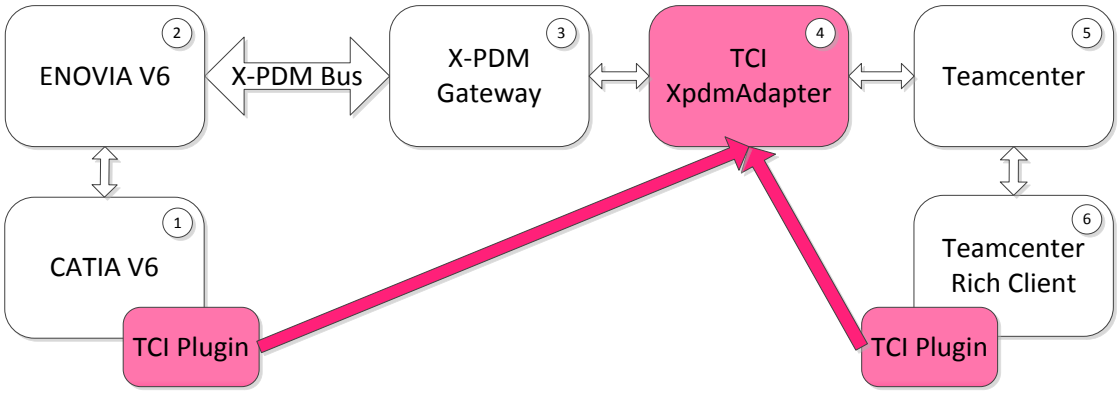

Figure 2-1 Infrastructure Overview

1. CATIA V6 Rich Client
2. ENOVIA V6 Server
3. X-PDM Gateway Instance
4. TCI XpdmAdapter
5. Teamcenter Server
6. Teamcenter Rich Client

2.2 Infrastructure Remarks

The TCI XpdmAdapter (4) must have full access to the X-PDM Gateway (3) shadow directory and should be hosted on the same system.

3 Installing the TCI XpdmAdapter

3.1 Prerequisites

3.1.1 Teamcenter

The Teamcenter 4-tier server has been installed.

The following values will be used in the following steps below:

Setting Name	Setting Value
Teamcenter URL	http://my.tc.server.url:8080/tc

Table 3.1: Teamcenter settings.

3.1.2 Teamcenter SOA Client

The Teamcenter SOA Client libraries of the file soa_client.zip of the Teamcenter installation package have to be extracted to a known directory, e.g. `C:\TC910\soa_client`.

3.1.3 ENOVIA

Before installing the Teamcenter Adapter the ENOVIA server and the required infrastructure must be installed as described in the ENOVIA documentation:

Installation and Administration | *ENOVIA* | *Unified Live Collaboration* | *X-PDM Gateway – Installation*

The following values will be used in the following steps below:

Setting Name	Setting Value
XPDM Gateway URL	http://my.gateway.url:8180/XPDMGateway
XPDM Gateway Site ID	XPDM
ENOVIA Gateway Site ID	V6

Table 3.2: ENOVIA settings.

3.2 Adapter setup

This section describes how you should install the TCI.

The adapter uses the X-PDM Infrastructure of the ENOVIA V6 to exchange data between ENOVIA and Teamcenter. As the adapter cannot function without the systems, both systems must be accessible from the adapter.

3.2.1 Setup as web application

The adapter can be deployed to a web application server.

The adapter will be installed in a directory on the host machine. This directory will be called installation directory in the following steps.

For example:

```
C:\XpdmAdapter
```

The web application server Tomcat has to be installed to a known directory, e.g. `C:\XpdmAdapter\apache-tomcat-7.0.57`.

The web application must have access to the Teamcenter File Client Cache (FCC) libraries in the classpath. Therefore the following lines have to be added to the Tomcat configuration file `[TOMCAT_HOME]/conf/catalina.properties`:

```
#
# Teamcenter environment configuration
shared.loader=[FMS_HOME]/jar/*.jar,[SOA_CLIENT]/java/libs/*.jar
```

For TC910 the setting `shared.loader` needs a different order:

```
shared.loader=[SOA_CLIENT]/java/libs/*.jar,[FMS_HOME]/jar/*.jar
```

Where `[FMS_HOME]` has to be replaced by the absolute or relative path to the `FMS_HOME` directory (Example: `C:/TC910/TCCS`).

And `[SOA_CLIENT]` is the location of the extracted Jar files from the Teamcenter SOA Client libraries (soa_client.zip) distributed with Teamcenter (see chapter 3.1.2).

The following environment variables have to be set for the web application:

Environment Variable	Example Value	Description
TCI_HOME	C:\XpdmAdapter	TCI Integration home directory
FMS_HOME	C:\TC910\TCCS	Teamcenter FMS home directory
JAVA_OPTS	%JAVA_OPTS% -Djlicman.platform="Windows_2003_amd64" -Djava.library.path=%FMS_HOME%;%FMS_HOME%/lib	Append additional Java options

Table 3.3: TCI web application environment variables

The following batch file can be used to start the Tomcat application server:

```
set FMS_HOME=<Teamcenter FMS Home directory>
```

```
set TCI_HOME=<TCI Integration Home directory>

set CATALINA_HOME=<Tomcat Catalina Home directory>

set JA-
VA_OPTS=%JAVA_OPTS% -Djlicman.platform="Windows_2003_amd64" -Djava.libra
ry.path=%FMS_HOME%;%FMS_HOME%/lib


call %CATALINA_HOME%\bin\startup.bat
```

The Adapter can now be deployed to the application server.

Copy the war file of the XpdmAdapter to the `webapps` directory of the web application server.

3.3 Adapter configuration

After the installation the TCI XpdmAdapter must be configured in the following configuration files in the installation directory.

3.3.1 Database configuration

The TCI makes use of a database to store and keep track of required data. The used database can be configured using the resource definition of the application server. The TCI will read the resource defined as "*jdbc/tciDB*".

The default resource can be overridden by defining the *PersistenceUnitName* in the settings (/XpdmAdapterSettings/Conversion/Database).

```
<Database>
        <PersistenceUnitName>jdbc/myTciDB</PersistenceUnitName>
 </Database>
```

Example resource definitions in Tomcat 7:

```
<Resource name="jdbc/tciDB" type="javax.sql.DataSource"
                auth="Container" driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
                maxActive="100" maxIdle="30" maxWait="10000"
                url="jdbc:derby:C:/TCI_HOME/tciDB;create=true;"
                username="tci"
                password="tci" />
<Resource name="jdbc/tciDB" type="javax.sql.DataSource"
                auth="Container" driverClassName="oracle.jdbc.OracleDriver"
                maxActive="20" maxIdle="10" maxWait="-1"
                url="jdbc:oracle:thin:@my.db-server-url.com:1521:orcl"
                username="tci"
                password="tci" />
```

If the resource is not defined a local and file based DerbyDB (JavaDB) will be created in the home directory.

This can be overridden by defining a custom database in the settings (/XpdmAdapterSettings/Conversion/Database).

```
<Database override="true" driverClassName="oracle.jdbc.OracleDriver"
url="jdbc:oracle:thin:@my.db-server-url.com:1521:orcl">
        <Property name="user" value="tci" />
```

```
        <Property name="password" value="tci" />
        <Property name="maxActive" value="20" />
        <Property name="maxIdle" value="10" />
        <Property name="maxWait" value="-1" />
</Database>
```

3.3.2 Logging configuration

The adapter will read and apply the log4j configuration from the following files in its base directory:

`log4j.xml`

`log4j.properties`

Copy the properties file `log4j.properties` of the installation package to the installation directory.

Please check the `*.File` entries of the properties file if they are pointing to a valid directory.

Further information about the configuration can be obtained from https://logging.apache.org/log4j/. If none of the files is present when starting the adapter it will apply its embedded default configuration, logging only to the standard output. To simplify the configuration a sample is distributed with the adapter.

3.3.3 General adapter configuration

Copy the files `XpdmAdapterSettings.xml` and `XpdmAdapterSettings.xsd` of the installation package to the installation directory.

The file `XpdmAdapterSettings.xml` defines all the functional settings for the adapter. The settings file can be evaluated using the `XpdmAdapterSettings.xsd` file delivered with the adapter.

It is divided into the three sections *Conversion*, *Teamcenter*, and *Enovia*.

In the next chapters the configuration of these sections will be described in details.

3.3.3.1 Conversion

Allows the configuration of properties for the conversion and mapping process of the data.

3.3.3.1.1 General Settings

<TempDir>C:\temp</TempDir>

Optional definition of a temporary directory to be used by the adapter. The adapter must have read and write access to this directory.

If the value is not defined the path "*<TCI Integration Home>\temp*" will be used.

<PersistentDir>C:\persistent</PersistentDir>

Optional definition of a directory to be used by the adapter to store persistent data. The adapter must have read and write access to this directory.

If the value is not defined the path "*<TCI Integration Home>\persistent*" will be used.

<Database>

Optional definition of the database to use. Please refer to section 3.3.1.

```
<RenameFileTypesAfterDownload enabled="true">
        <FileType>CATDrawing</FileType>
</RenameFileTypesAfterDownload>
```

List of file types which will be renamed to `obid.extension`, to support multiple objects with the same File name. Should only be used with file types which are not referenced by other files (CATDrawing, JT etc.).

3.3.3.1.2 Custom Attribute Mapping

```
<CustomAttributeMapping    enabled="true">
        <CustomObject      xname="Product"
                           tctype="ItemRevision"
                           mappingtype="DefaultCustomization">
                <CustomAttribute    xname="Description"
                                    tcname="object_desc"
                                    type="string"
                                    export="true"
                                    import="false"/>
                <CustomFormAttributes    formtype="ItemRevision Master"
                                         relationname="IMAN_master_form_rev">
                        <CustomAttribute    xname="Supplier Name"
                                            tcname="user_data_1"
                                            type="string"
                                            export="true"
                                            import="true"/>
                </CustomFormAttributes>
        </CustomObject>
</CustomAttributeMapping>
```

Definition of the custom attribute mapping.

Each element defines the *xname* and the *tcname*.

The *xname* defines the name of the XPDMXML custom property to use.

The *tcname* defines the name of the Teamcenter property to use.

<CustomAttributeMapping>

The mapping can be switched on or off using the *enabled* attribute.

It can contain multiple *CustomObject* definitions.

<CustomObject>

Each object definition can contain multiple *CustomAttribute* and *CustomFormAttributes* definitions.

<CustomAttribute>

The *type* defines the data type of the property to use.

The direction of the mappings is defined independently by the *import*/*export* attributes for each *CustomAttribute*. If *import* is true the attribute will be imported to Teamcenter. If *export* is true the attribute will be exported from Teamcenter.

The optional integer attribute *maxlength* defines the maximum length of the attribute. If the maximum length of the attribute is defined, the *truncate* attribute flag enables the truncate functionality

which will truncate the attribute to the maximum length. This is checked if the setting *AttributeCheck* *(see chapter 3.3.3.1.3)* is enabled.

<CustomFormAttributes>

The *CustomFormAttributes* tag can be used to refer not the Teamcenter object defined in the CustomAttribute but to a *formtype* attached to in with the specified *relationname*.

It can contain multiple *CustomAttribute* definitions.

3.3.3.1.3 XpdmXml Export Checks definitions

```
<XpdmXmlExportChecks>
        <AttributeCheck  enabled="true">
                <XType xname ="Product">
                        <XAttribute     xname="Description"
                                        mappingtype="DefaultCustomization"
                                        maxlength="60"
                                        truncate="true"/>
                </XType>
        </AttributeCheck>
        <InstanceNameCheck enabled="true"/>
        <FileRefCheck enabled="false"/>
        <FileNameCheck enabled="false"/>
        <ProductLeafRepresentationCheck enabled="false" onlyWarning="true"/>
        <RepNameCheck enabled="true"/>
</XpdmXmlExportChecks>
```

Definition of the export checks.

Each check must be enabled and disabled with the *enabled* attribute. The optional *onlyWarning* attribute can be set to true. Then the check is handled as warning condition not as error condition.

The *AttributeCheck* is used to check the *maxlength* attribute in the *CustomAttribute* *(see chapter 3.3.3.1.2)* setting.

Additionally the standard attribute length can be checked. The example above shows the limitation of the Description to maximum length of 60. If the maximum length of the attribute is defined, the *truncate* attribute flag enables the truncate functionality which will truncate the attribute to the maximum length.

The *InstanceNameCheck* checks the uniqueness of instance name per parent object.

The *FileRefCheck* checks the multiple use of a File Ref from different Objects.

The *FileNameCheck* checks the uniqueness of file names in the files section.

The *ProductLeafRepresentationCheck* checks the existence of an attached Representation at a Product node, if the Product node is a leaf node.

The *RepNameCheck* checks for duplicate representation names. ID Attribute in Rep3D, Drawing, Document, and GenericDocument Object.

3.3.3.1.4 Teamcenter Project definitions

```
<Projects>
        <DefaultProject name="Default">
                <ProjectName>Standard</ProjectName>
                <Organization>Company Name</Organization>
                <Owner>DemoDesigner</Owner>
                <LifeCycle>VPLM_SMB</LifeCycle>
```

```
            <Maturity>SHARED</Maturity>
        </DefaultProject>
        <Project name="test">
            <ProjectName>Standard</ProjectName>
            <Organization></Organization>
            <Owner></Owner>
            <LifeCycle>VPLM_SMB</LifeCycle>
            <Maturity>SHARED</Maturity>
        </Project>
</Projects>
```

Definition of projects and the corresponding values for *Project*, *Organization*, *Owner*, *LifeCycle* and *Maturity*. The *Projects* section must include one *DefaultProject* that will be used when no other defined project matches.

The defined values for the projects define which values will be set to the items when sending them to ENOVIA. If no project is defined or no values are defined for a project the values from the default project are used.

The values for *Organization* and *Owner* can be left blank. These will be set to the values to the importing administration user on the ENOVIA import.

For the Non CATIA document import the Maturities and LifeCycle can be defined for Document and GenericDocument types.

The Defaults for Document and GenericDocument types are:

Default Document Maturity: *Exists*
Default Document Lifecycle: *Document*
Default GenericDocument Maturity: *WIP*
Default GenericDocument Lifecycle: *Controlled Production Release Rev2*

These defaults can be set in the Project settings:

```
<DefaultProject name="Default">
        …
        <DocumentLifeCycle>Document</DocumentLifeCycle>
        <DocumentMaturity>WIP</DocumentMaturity>
        <GeneralDocumentLifeCycle>Controlled Production Release Rev2</GeneralDocumentLifeCycle>
        <GeneralDocumentMaturity>Exists</GeneralDocumentMaturity>
</DefaultProject>
```

3.3.3.1.5 File type mapping definitions

3.3.3.1.5.1 V6 to Teamcenter file type mapping

```
<V6TcFileType>
        <V6FileFormat v6format = "CATPart">
            <Main>
                <TCDSType>CMI2Part</TCDSType>
                <TCDSRelation>IMAN_specification</TCDSRelation>
                <TCNRType>CATPart</TCNRType>
            </Main>
            <Aux>
                <TCDSType>CMI2AuxPart</TCDSType>
                <TCDSRelation>IMAN_specification</TCDSRelation>
                <TCNRType>CATPart</TCNRType>
            </Aux>
        </V6FileFormat>
```

</V6TcFileType>

The *V6TcFileType* can contain multiple *V6FileFormat*s defining how the specified file should be imported to Teamcenter.

Each *V6FileFormat* defines a *TCDSType*, a *TCDSRelation*, and a *TCNRType* in the *Main* section. The same is defined for *Aux* datasets. The *Aux* definition is used for shared representations.

3.3.3.1.5.2 Teamcenter to V6 file type mapping

```
<TcV6FileType>
        <V6Representation fileextension="CATPart" dstype="CMI2Part">
                <RepresentationType>Rep3D</RepresentationType>
                <DerivedType>DirectModel</DerivedType>
        </V6Representation>
</TcV6FileType>
```

The *TcV6FileType* can contain multiple *V6Representation*s defining how the specified file should be exported from Teamcenter.

Each *V6Representation* defines a *RepresentationType* and a *DerivedType*.

3.3.3.1.5.3 Customization registration

```
<Customizations>
        <CustomizationPoint
                custpoint="com.tsystems.tci.customization.CustomizationPointX"
                entrypoint="my.customization.MyCustomizationPointImpl"/>
</Customizations>
```

The behavior of the integration can be modified using customization points. The *Customizations* tag may contain multiple *CustomizationPoint*s defining which customization implementation should be used. Each *CustomizationPoint* defines a *custpoint* to be overridden and an *entrypoint* specifying the overriding implementation.

It is not necessary to set a customization point. All customization points are implemented using the integration default behavior.

If you require any kind of customization please contact the support team.

3.3.3.1.5.4 SetTimeModified setting

```
<SetTimeModified>
        <Rep3D disabled="true"/>
        <Drawing disabled="true"/>
        <Document disabled="true"/>
        <GenericDocument disabled="true"/>
</SetTimeModified>
```

The *<TimeModified>* tag is by default written to the *<Rep3D>*, *<Rep3DAggr>*, *<Drawing>*, *<DrawingAggr>*, *<Document>*, and *<GenericDocument>* tags in the Metadata.xml import file. With this Time information the import can skip already imported files and optimize the performance of the import process.

This optimization can be disabled, e.g. to resolve multi model links in several contexts.

3.3.3.1.5.5 SharedFileTypeExtensions setting

```
<SharedFileTypeExtensions>
        <SharedFileTypeExtension>model</SharedFileTypeExtension>
        <SharedFileTypeExtension>cgr</SharedFileTypeExtension>
        <SharedFileTypeExtension>CATDrawing</SharedFileTypeExtension>
</SharedFileTypeExtensions>
```

Datasets are by default written as *Rep3DAggr* or *DrawingAggr* object in the metadata xml input file. All File extensions which are defined in the *SharedFileTypeExtension*s setting are written as Shared object type *Rep3DInst/Rep3D* or *DrawingInst/Drawing* objects in the metadata xml input file and they will be created as a shared object in the ENOVIA V6.

3.3.3.1.5.6 ExclusionListFile setting

<ExclusionListFile>C:\tmp\exclusionlist.xml</ExclusionListFile>
The *ExclusionListFile* is the exclusion list with the invalid CAD objects which should not be imported to V6.

Example for an Exclusion list settings file.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<EL>
      <ItemRev>
            <Id>DREP2-TSI2-0001</Id>
            <Type>Design Revision</Type>
            <Revision>1</Revision>
      </ItemRev>
      <ItemRev>
            <Id>DREP2-TSI2-0001</Id>
            <Type>Design Revision</Type>
            <Revision>2</Revision>
      </ItemRev>
</EL>
```

3.3.3.1.5.7 V5ToV6Preprocessing setting

<V5ToV6Preprocessing enabled="true">
<V5PreprocessingCommand>C:\CATUpdate\go\catupdate.bat C:\CATUpdate\go\TCI.opt</V5PreprocessingCommand>
</V5ToV6Preprocessing>
The Functionality for the V5 Preprocessing with CATUpdate must be enabled to use the CATUpdate tool from T-Systems.

3.3.3.1.5.8 MaturityStatusMappingList setting

<MaturityStatusMappingList enabled="true">
<MaturityStatusMapping v6maturity="SHARED" tcstatus="TCM Released"/>
</MaturityStatusMappingList>
The maturity status mapping list defines the mapping between the V6 Maturity and the Status object name in Teamcenter. When the functionality is enabled all V6 Objects with the Maturity defined in the *v6maturity* attribute will get a Status object with the name from the *tcstatus* attribute.

3.3.3.1.5.9 StatusUpdateHandling setting

<StatusUpdateHandling enabled="true">
<AdditionalStatusUpdate sourceType="Design Revision" targetRelationType="IMAN_master_form" targetObjectType="Design Revision Master"/>
<AdditionalStatusUpdate sourceType="Design Revision" targetRelationAttribute="structure_revisions" targetObjectType="BOMView Revision"/>
</StatusUpdateHandling>
Used to add Additional objects for the Status update process.

The *sourceType* defines the Item Revision type under which the additional objects should be updated.

The example settings show an example for a Relation based setting (IMAN_master_form) and a Attribute based setting (structure_revisions).

3.3.3.2 Teamcenter

Allows the configuration of Teamcenter related information like login information, server and dispatcher configuration.

3.3.3.2.1 Teamcenter Server Definition

```
<Server>
        <Url>http://my.tc.server.url:8080/tc</Url>
        <Username>myXpdmAdapterUser</Username>
        <Password plainText="true">myXpdmAdapterPassword</Password>
        <Group></Group>
        <Role></Role>
</Server>
```

Definition of the Teamcenter server and the login credentials of the XpdmAdapter user.

The *Username* and the *Password* belong to an existing Teamcenter user. You can also specify the *Group* and *Role* in which the user will act.

3.3.3.2.2 Teamcenter Server Password

The password in the "`XpdmAdapterSettings.xml` (//XpdmAdapterSettings /Teamcenter/Server/Password)" can be stored as plain text or encrypted.

```
<Password plainText="true">myXpdmAdapterPassword</Password>
```

To encrypt the password for the adapter the following script can be used:

```
set TCI_HOME=<TCI Integration Home directory>
set CATALINA_HOME=<Tomcat Catalina Home directory>
set WEB_APP_NAME=XpdmAdapter

set JAVA_OPTS=%JAVA_OPTS% -Djlicman.platform="Windows_2003_amd64"
-Djava.library.path=%FMS_HOME%;%FMS_HOME%/lib
set CLASSPATH=%CATALINA_HOME%\webapps\%WEB_APP_NAME%\WEB-
INF\lib\*;%CATALINA_HOME%\webapps\%WEB_APP_NAME%\WEB-INF\classes

java com.tsystems.tci.xtgw.Main --encryptPassword
```

The adapter will prompt the user to enter the password. The password will only be written out to the standard output and must be copied to the configuration file manually.

The command will generate a file named "`passwordKey`" and the encrypted password, that is printed to the command line.

Note:

If the "`passwordKey`" file exists it will not be overwritten.

The encrypted password may differ after each execution of the encryption. It can be set as follows.

<Password plainText="false">INdlaJZMaTQO0MblX+/C1Q==</Password>

3.3.3.2.3 Other Teamcenter Settings

<PolicyPath>TCI_Policy</PolicyPath>

Optional definition of the Teamcenter policy to load.

<GetPropertiesChunkSize>1000</GetPropertiesChunkSize>

<ExpandChunkSize>1000</ExpandChunkSize>

Optional definition of the maximal chunk sizes to use when requesting metadata from the Teamcenter server.

Warning: Modifying this settings can affect the performance of the tool and die utilization of the Teamcenter server.

<FileUploadChunkSize>1</FileUploadChunkSize>

<FileDownloadChunkSize>1</FileDownloadChunkSize>

Optional definition of the chunk sizes to use to up- and download files using the Teamcenter FCS.

<ReallyRemoveExistingNotProvided enabled="true" />

Setting whether to remove unknown occurrences in Teamcenter or not.

<CreateItemClass>Item</CreateItemClass>

Name of the Teamcenter Item type to use. The default Item class is “Item”.

<WorkingGroup>V6Group</WorkingGroup>

Name of the adapter working group in Teamcenter. *ItemRevisions* owned by ENOVIA will be moved to this group.

```
<Dispatchers>
        <Dispatcher enabled="false">
                <FileType>CATPart</FileType>
                <Provider>TSYSTEMS</Provider>
                <Priority>3</Priority>
                <Servicename>cmitojt</Servicename>
                <Type>ONDEMAND</Type>
        </Dispatcher>
</Dispatchers>
```

Definitions for optional dispatchers to use. Each dispatcher can be enabled separately.

```
<DatasetRelations>
        <Relation>IMAN_specification</Relation>
        <Relation>IMAN_Rendering</Relation>
</DatasetRelations>
```

Defines the relation types to be followed to datasets when the export functionality of the adapter is used.

```
<ReportReceiverList enabled="false">
        <Receiver>myAdapterUser1</Receiver>
        <Receiver>infodba</Receiver>
</ReportReceiverList>
```

Defines a list of Teamcenter users to receive reports on adapter actions. The reports will be delivered through Teamcenter to the Mailbox of the defined users.

The feature can only be used with Teamcenter 9.1 or newer.

```
<LinkedDrawingHandling enabled="true">
        <LinkedDrawingDatasetType>CATDrawing</LinkedDrawingDatasetType>
</LinkedDrawingHandling>
```

Defines a list of Teamcenter Drawing Dataset types which should be searched as referenced at ItemRevisions. The TCIC Integration uses links for the Drawing handling in Teamcenter. If the *LinkedDrawingHandling* is enabled the Integration searches for referenced Drawings and transfers these drawings to ENOVIA.

```
<ItemQuerySettings>
        <ItemRevisionTypes>
                <ItemRevisionType></ItemRevisionType>
                <ItemRevisionType>ItemRevision</ItemRevisionType>
                <ItemRevisionType>Design Revision</ItemRevisionType>
                <ItemRevisionType>Part Revision</ItemRevisionType>
        </ItemRevisionTypes>
        <ItemRevisionAttributeNames>
                <ItemRevisionAttributeName
                        display="Item ID">Item ID</ItemRevisionAttributeName>
                <ItemRevisionAttributeName
                        display="Item Revision">Revision</ItemRevisionAttributeName>
                <ItemRevisionAttributeName
                        display="Name">Name</ItemRevisionAttributeName>
                <ItemRevisionAttributeName
                        display="Description">Description</ItemRevisionAttributeName>
        </ItemRevisionAttributeNames>
        <ResultAttributeNames>
                <ResultAttributeName>item_id</ResultAttributeName>
                <ResultAttributeName>item_revision_id</ResultAttributeName>
                <ResultAttributeName>owning_user</ResultAttributeName>
                <ResultAttributeName>owning_group</ResultAttributeName>
                <ResultAttributeName>last_mod_date</ResultAttributeName>
        </ResultAttributeNames>
        <MaxResultToReturn>25</MaxResultToReturn>
</ItemQuerySettings>
```

ItemRevisionTypes defines the settings for the Teamcenter Item Query.

Use empty *ItemRevisionType* for search all types.

The *ItemRevisionAttributeName* must be the "User Entry Name" in the Query definition of the Teamcenter Query Builder application.
Example:

The "*item_id*" attribute "Item ID"

If the *ResultAttributeNames* tag does not exist, then all not empty attributes are returned.

3.3.3.3 Enovia

<GatewayUrl>http://my.gateway.url:8180/XPDMGateway/webservice?id=XPGEventsService</GatewayUrl>
Definition of the gateway server URL to send and receive events on.

<MonitoringUrl>http://my.gateway.url:8180/XPDMGateway/webservice?id=XPGMonitoringService</MonitoringUrl>
Definition of the monitoring server URL (normally the same as the gateway server).

<ReceiveInterval>5000</ReceiveInterval>
Optional definition in which interval the events should be received in milliseconds (ms).

<ShadowPath>C:\enovia\XPDMGateway\shadow</ShadowPath>
Definition of the shadow directory of the X-PDM gateway.

<SourceSiteId>XPDM</SourceSiteId>
Definition of the X-PDM gateway source site id (optional).

<TargetSiteId>V6</TargetSiteId>
Definition of the X-PDM gateway target site id.

<SchemaPath>C:\enovia\XPDMGateway\STAGING\ematrix\WEB-INF\resources\xsd\XPG\XPDMXML.xsd</SchemaPath>
Definition of the schema to use to validate X-PDM messages (can be found in the X-PDM Gateway). If the file path is not defined or incorrect, the incoming and outgoing XPDMXML files will not be validated but processed.

<TransitionEngineLogPath>C:\Temp</TransitionEngineLogPath>
The *TransitionEngineLogPath* must be set to the path to the XPG Client Service Temp directory.

The defined directory must be accessible for the adapter and the used XPGClientService must be configured to log the directory.

The XPG ClientService must run **without** the option "*-deleteLocalLogs*" for the collection of result data from the transition logs!

For more information about the XPGClientService please refer to the ENOVIA V6 documentation:

Installation and Administration | *ENOVIA* | *Unified Live Collaboration* | *X-PDM Gateway - Installation* | *Setting Up the VPM V6 Connector on V6 Client* | *Configuring XPGClient Services*

<FeedbackTryInterval>6000</FeedbackTryInterval>
Definition in which interval the feedback on sent events should be monitored in milliseconds (ms).

<MaxFeedbackTry>100</MaxFeedbackTry>
Definition how often the feedback on events should be monitored.

<Customization>DefaultCustomization</Customization>
Definition of the customization to use in X-PDM events. This customization must be defined in the `pdm-cust-config.xml` in the ENOVIA installation.

3.3.4 Event Configuration

The TCI XpdmAdapter utilizes the ENOVIA X-PDM Gateway and comes with its own custom events. These must be added to the event configuration in the X-PDM Gateway:

Add *<DomainImport ref="CMI"/>* to the `event-config.xml` file in the directory `[GATEWAY_HOME]\config\event`.
Create the folder `[GATEWAY_HOME]\config\event\CMI` and the file `event-config.xml` in the new folder:

```
<?xml version="1.0" encoding="UTF-8"?>
<EventConfig>
        <Event Name="CMIExportV6FromTc"
                Type="CMI"
                ProcessingLocations="LocalOnly"
                HasMetaData="False"
                HasBulkData ="False" />
</EventConfig>
```

For more information about the event usage of the adapter please refer to chapter A of this document.

For more information about the event configuration please refer to the ENOVIA V6 documentation:

Installation and Administration | *ENOVIA* | *Unified Live Collaboration* | *X-PDM Gateway - Installation* | *Setting Up the Gateway Server For ENOVIA* | *Managing Gateway Server* | *Event Configuration*

3.3.5 V6 Environment File

After the installation of the ENOVIA X-PDM Gateway the environment variable ENABLE_DETAILED_REPORT=1 has to be set in the environment file used by the ENOVIA XPGClientService.

For more information about the environment configuration please refer to the ENOVIA V6 documentation:

Installation and Administration | *CATIA, DELMIA, SIMULIA* | *Installation and Configuration* | *Configuring the Software* | *Configuring Environments* | *What Is An Environment?*

3.3.6 ENOVIA Custom Configuration

The PDM custom configuration (`pdm-cust-config.xml`) of ENOVIA must contain the following attributes.

```
<!-- PLMProductDS -->
<Object       XPGXMLType="Product"
              V6Type="PLMProductDS"
              MappingType="DefaultCustomization">
    <attribute    XPGName="V_usage"
                  V6Name="V_usage"
                  type="string"
                  Export="true"
                  Import="true"/>
</Object>
<Object       XPGXMLType="ProductInst"
              V6Type="PLMProductInstanceDS"
              MappingType="DefaultCustomization"/>
<Object       XPGXMLType="Rep3DAggr"
              V6Type="PLMRepresentationDS"
              MappingType="DefaultCustomization">
    <attribute    XPGName="Representation Kind"
                  V6Name="V_RepKind"
                  type="string"
                  Export="true"
                  Import="true"/>
    <attribute    XPGName="V_discipline"
                  V6Name="V_discipline"
                  type="string"
                  Export="true"
                  Import="true"/>
    <attribute    XPGName="V_usage"
                  V6Name="V_usage"
                  type="string"
                  Export="true"
                  Import="true"/>
</Object>
<Object       XPGXMLType="Rep3D"
              V6Type="PLMRepresentationDS"
              MappingType="DefaultCustomization">
    <attribute    XPGName="Representation Kind"
                  V6Name="V_RepKind"
                  type="string"
                  Export="true"
                  Import="true"/>
    <attribute    XPGName="V_discipline"
                  V6Name="V_discipline"
                  type="string"
```

```
                    Export="true"
                    Import="true"/>
        <attribute  XPGName="V_usage"
                    V6Name="V_usage"
                    type="string"
                    Export="true"
                    Import="true"/>
</Object>
<Object     XPGXMLType="Rep3DInst"
            V6Type="PLMRepresentationInstanceDS"
            MappingType="DefaultCustomization"/>
<Object     XPGXMLType="DrawingAggr"
            V6Type="PLMRepresentationDS"
            MappingType="DefaultCustomization">
        <attribute  XPGName="Representation Kind"
                    V6Name="V_RepKind"
                    type="string"
                    Export="true"
                    Import="true"/>
</Object>
<Object     XPGXMLType="Drawing"
            V6Type="PLMRepresentationDS"
            MappingType="DefaultCustomization">
        <attribute  XPGName="Representation Kind"
                    V6Name="V_RepKind"
                    type="string"
                    Export="true"
                    Import="true"/>
</Object>
<Object     XPGXMLType="DrawingInst"
            V6Type="PLMRepresentationInstanceDS"
            MappingType="DefaultCustomization"/>
```

For more information about the ENOVIA custom configuration please refer to the ENOVIA V6 documentation:

Installation and Administration | ENOVIA | Unified Live Collaboration | X-PDM Gateway - Administration | Configuring The Data Model Mapping | pdm-cust-config.xml Configuration File

3.4 Trouble shooting

During startup of the TCI XpdmAdapter you may get an exception like this:

```
java.io.IOException: Library not found: Windows_Server_2008_R2_amd64_6_1
        at com.tsystems.cadcam.jlicman.tools.F.<clinit>(Unknown Source)
        at com.tsystems.cadcam.jlicman.LicenseManager.<clinit>(Unknown Source)
        at com.tsystems.xpg.XPGAdapter.run(Unknown Source)
        at java.lang.Thread.run(Unknown Source)
```

Where Windows_Server_2008_R2_amd64_6_1 describes your current operating system and the Java bit-mode.

Solution:

Use one of the following commands to start up the TCI XpdmAdapter:

- 64 bit Java runtime:
  java -Djlicman.platform="Windows_2003_amd64" -jar adapter.jar
- 32 bit Java runtime:
  java -Djlicman.platform="Windows_2003" -jar adapter.jar

4 TcIC compatibility

To support as most as possible environments the adapter can be configured to run in TcIC compatibility mode. This will modify the default behavior to match some specialties of the environment.

To switch on the TcIC compatibility mode the following configurations must be set:

1. In the adapter configuration file:

```
<Teamcenter>
        …
        <TcICCompatibility enabled="true"/>
        …
</Teamcenter>
```

2. In the Teamcenter Rich Client environment:

   TCI_GATEWAY_TCIC_COMPATIBILITY=true

5 Installing the license manager

5.1 Remarks

The TCI XpdmAdapter requires licman20 as its license manager.

Licman20ptf19 is recommended.

For the installation of Licman please refer to the Licman documentation.

http://cmi-support.com/ftp_pub/download/licman/licman20.pdf

5.2 Windows 7

During the installation of the Licman, you have to specify a temporary working directory. Choose a directory where the user has full access rights. With Licman20 prior to ptf19, the windows/temp directory that is given as a default will not work under Windows 7.

5.3 Running Licman as a regular executable on Windows

If you install Licman with the setup program on Windows, it will be run as a service and require administrator rights. Beginning with ptf19, it is possible to launch Licman as a regular executable when the TCI XpdmAdapter is started.

You need to set the license path variable:

SET LICMAN_LICENSE_PATH=<port>@<host>

Then call licman20_lld.exe as a regular executable.

No execution of the setup routine is required in this case.

For more information, see *Licman User Manual*.

6 Installing the TCI Teamcenter plug-in

6.1 Installation

The following file has to be copied to the Teamcenter client installation. The variable %TC_DIR% is the Teamcenter installation directory.

Copy	to
com.tsystems.tci.xpg_tc-[version].jar	%TC_DIR%\portal\plugins

Table 6.1: TCI Teamcenter plug-in file to be installed

The following environment variables must be set for the plug-in:

Environment Variable	Example Value	Default Value
TCI_SOURCE_GATEWAY_SITE_ID	XPDM	<must be set>
TCI_TARGET_GATEWAY_SITE_ID	V6	<should be set; if not set then the value from XPG_XPDM_SOURCE_GATEWAY_SITE_ID is used>
TCI_EVENT_NAME	PRODUCTMirror	CMIExportV6FromTc
TCI_GATEWAY_URL	http://my.gateway.url:8180/XPDMGateway	<must be set>
TCI_GATEWAY_USER	myXpdmAdapterUser	<must be set>
TCI_GATEWAY_QUEUE_NAME	Tc_1_{user}_{uuid}	tc_{user}
TCI_SEND_PSE_RENDERING_ENABLED	true	true
TCI_GATEWAY_TCIC_COMPATIBILITY	false	false
TCI_EXPORT_TYPE	V6ExportFile	V6ExportFile
TCI_EXPORT_NAMED_REF	V6CMI	V6CMI

Table 6.2: Teamcenter plug-in - environment variables

The queue name set in TCI_GATEWAY_QUEUE_NAME can contain the following placeholders, that will be replaced for every event.

- {user} User name of the current Teamcenter user.
- {uuid} A random generated UUID.
- {event} Type of the event.

7 Installing the CATIA V6 client extensions

Beside the XpdmAdapter, TCI adds some features to the CATIA V6 client. This section describes how to install the client extensions.

1. Copy the client installation package TCI_<xx> (xx stands for the TCI Release) to your installation directory, like `C:\Program Files\T-Systems`:

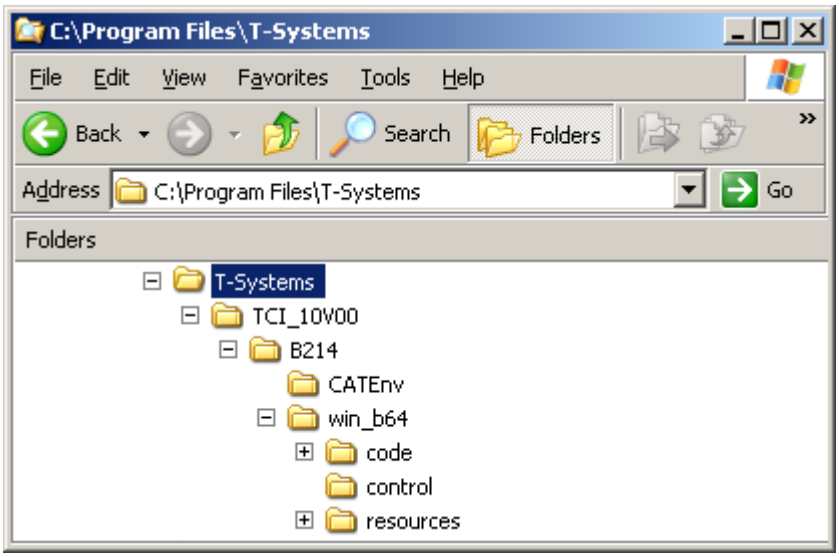

Figure 7-1 Installation directory of the CATIA V6 client extensions

2. Create a new environment file for CATIA V6 client with TCI extensions.

   To create the new environment file use the native Dassault Systèmes utility `setcatenv` from the CATIA V6 client installation, default:
   `C:\Program Files\Dassault Systemes\B214\win_b64\code\bin\setcatenv.exe`

   To create the new environment file you will need the existing/currently used environment from which you want to create the new environment, default:
   `C:\Program Files\Dassault Systemes\B214\CATEnv\Env.txt`

   Open a Windows console and run `setcatenv` like follows:

```
C:\> "C:\<…>\setcatenv.exe" -d "C:\<…>\T-Systems\TCI_10V00\B214\CATEnv" -e
TCIEnv -p "C:\<…>\T-Systems\TCI_10V00\B214" -cf "C:\<…>\Dassault
Systemes\B214\CATEnv\Env.txt" -icon no -desktop no -menu no
```

   For more information about creating a new environment please refer to ENOVIA V6 documentation:
   Installation and Administration | *CATIA, DELMIA, SIMULIA* | *Installation and Configuration* | *Configuring the Software* | *Configuring Environments*

   After executing `setcatenv` the new environment file is available:
   `C:\Program Files\T-Systems\TCI_10V00\B214\CATEnv\TCIEnv.txt`

3. Create a startup script to launch the CATIA V6 client:

   Create an empty file `C:\Program Files\T-Systems\TCI_10V00\TCIStart.bat` and fill in the following settings/commands taking your paths into account.

```
:: ########### TCIStart.bat ###### begin ########
set PATH=C:\Program Files\Dassault Systemes\B214\win_b64\code\bin;%PATH%
set TCI_ENVDIR=C:\Program Files\T-Systems\TCI_10V00\B214\CATEnv

:: ----------------------------------------
:: Additional TCI Environment Variables START
:: ----------------------------------------
:: ### TCI Debug setting ###
rem set TCI_DEBUG=ON

:: ### ENOVIA V6 output ###
rem set CNEXTOUTPUT=console or C:\temp\catia.txt

:: ----------------------------------------
:: Additional TCI Environment Variables END
:: ----------------------------------------

start CATSTART -run CNEXT -env TCIEnv -direnv "%TCI_ENVDIR%" -nowindow

:: ########### TCIStart.bat ###### end ########
```

   You can add additional TCI client environment settings into the start script.

7.1 Client extensions configuration

The client must be configured using the CATIA options dialog.

The **TCI** configuration dialog under **Options | General | Compatibility** shows some version information of the integration.

Additionally the *Server Url* can be configured to connect to the TCI integration server. The client supports the encryption of the communication channel using SSL. To enable this feature the client must be configured to connect a https Url.

Examples:

http://10.0.0.100/XpdmAdapter

http://my.tci-integration-server.com/XpdmAdapter

https://my.tci-integration-server.com/XpdmAdapter

To verify the connectivity to the configured server the *Check Server* button can be used. A message will be shown to indicate successful or unsuccessful connections.

A Event flow

The adapter utilizes the ENOVIA X-PDM infrastructure to communicate.

Queue names

The adapter can be configured to use different queue names to send the event. There are three possibilities to configure the queue names that have different behavioral results.

- Constant queue name

  All events are sent in the same queue. The evens are processed in the order as they have entered the queue. If an event fails the whole queue will be blocked and no other events can be processed.

- User based queue name

  All events sent by a single user are sent in the same queue. The order of the events sent by the same user is preserved. If an event fails the user queue will be blocked but other user queues can still be processed.

- Event based queue name

  Every event will get its own queue on the gateway. The order of the events cannot be guaranteed.

Interactive send event from Teamcenter Rich Client

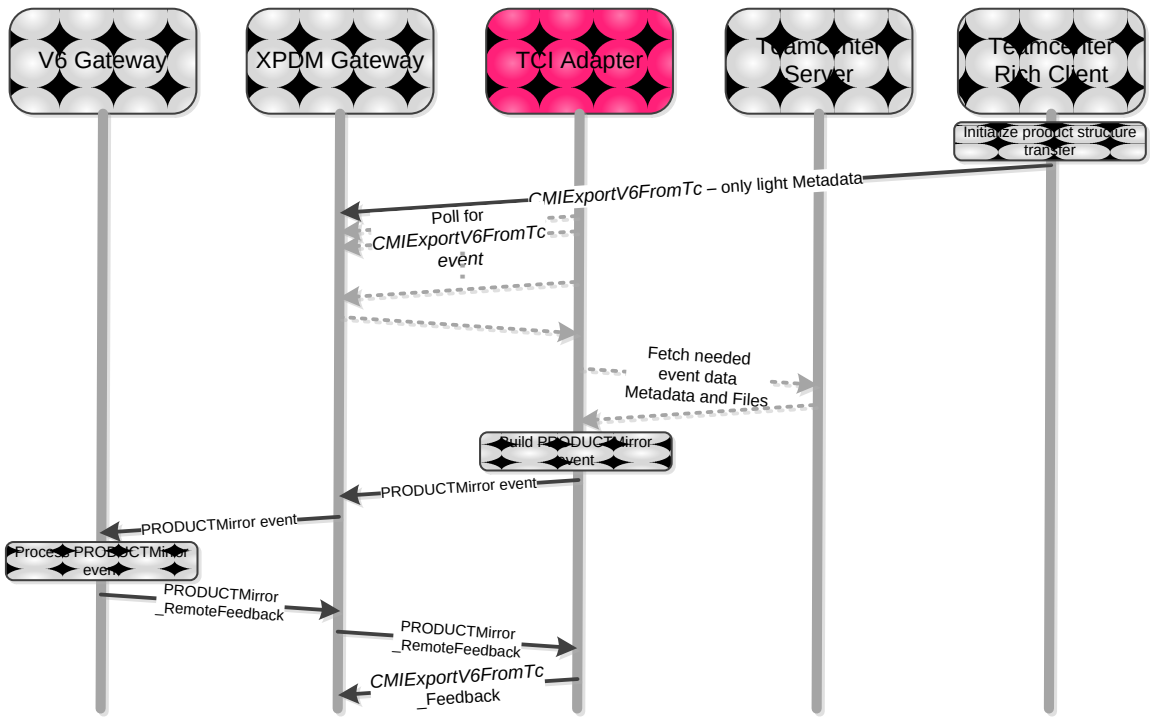

Figure A.1: Interactive send – event flow

The Teamcenter Rich Client sends its own event to the X-PDM Gateway. The CMIExportV6FromTc event is a local event on the X-PDM Gateway. This means that the source and target of the event is the X-PDM Gateway and the event is not distributed to the V6 Gateway. The event builds a bracket around the generation and processing of the PRODUCTMirror event.

List of Sources

Installation and Administration | ENOVIA | Unified Live Collaboration | X-PDM Gateway – Installation

Installation and Administration | ENOVIA | Unified Live Collaboration | X-PDM Gateway – Installation | Setting Up the VPM V6 Connector on V6 Client | Configuring XPGClient Services

Installation and Administration | ENOVIA | Unified Live Collaboration | X-PDM Gateway – Installation | Setting Up the Gateway Server For ENOVIA | Managing Gateway Server | Event Configuration

Installation and Administration | CATIA, DELMIA, SIMULIA | Installation and Configuration | Configuring the Software | Configuring Environments | What Is An Environment?

Installation and Administration | CATIA, DELMIA, SIMULIA | Installation and Configuration | Configuring the Software | Configuring Environments

Installation and Administration | ENOVIA | Unified Live Collaboration | X-PDM Gateway – Administration | Configuring The Data Model Mapping | pdm-cust-config.xml Configuration File

Change History / Release Notes

Version	Last revised	Author/Editor	Changes/Comments
1.0.0	2012-09-21	Christoph Schlameuß	Initial version
1.0.2	2012-12-06	Christoph Schlameuß	Updated documentation to adapter version 1.0.2
1.1.0	2013-03-01	Christoph Schlameuß	Updated documentation to adapter version 1.1.0
1.2.0	2013-05-24	Christoph Schlameuß	Updated documentation to adapter version 1.2.0
2.0.0	2013-09-27	Christoph Schlameuß	Updated documentation to adapter version 2.0.0
2.0.1	2013-11-08	Ulrich Mehringer	Review and enhancements
2.2	2015-01-20	Ulrich Mehringer	Review